



Understanding the Front-end of Large-scale Engineering Programs

Lucae, Sebastian; Rebentisch, Eric ; Oehmen, Josef

Published in:
Procedia Computer Science

Link to article, DOI:
[10.1016/j.procs.2014.03.079](https://doi.org/10.1016/j.procs.2014.03.079)

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Lucae, S., Rebentisch, E., & Oehmen, J. (2014). Understanding the Front-end of Large-scale Engineering Programs. *Procedia Computer Science*, 28, 653–662. <https://doi.org/10.1016/j.procs.2014.03.079>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Conference on Systems Engineering Research (CSER 2014)

Eds.: Azad M. Madni, University of Southern California; Barry Boehm, University of Southern California;
Michael Sievers, Jet Propulsion Laboratory; Marilee Wheaton, The Aerospace Corporation
Redondo Beach, CA, March 21-22, 2014

Understanding the Front-end of Large-Scale Engineering Programs

Sebastian Lucae^a, Eric Rebentisch^b, Josef Oehmen^{c*}

^a*Technische Universität München (TUM), Munich, Germany*

^b*Massachusetts Institute of Technology (MIT), Cambridge, MA, USA*

^c*Technical University of Denmark (DTU), Lyngby, Denmark*

Abstract

Large engineering programs like sociotechnical infrastructure constructions of airports, plant constructions, or the development of radically innovative, high-tech industrial products such as electric vehicles or aircraft are affected by a number of serious risks, and subsequently commonly suffer from large cost overruns. Significant problems in program execution can be traced back to practices performed, or more frequently not performed, in the so-called “fuzzy front end” of the program. The lack of sufficient and effective efforts in the early stages of a program can result in unstable, unclear and incomplete requirements, unclear roles and responsibilities within the program organization, insufficient planning, and unproductive tensions between program management and systems engineering.

This study intends to clarify the importance of up-front planning to improve program performance, to propose a model for the front-end of large-scale engineering programs based on a review of existing, suitable models in literature and to better understand the complexity drivers that are impeding reliable planning and common planning mistakes made in large-scale engineering programs.

© 2014 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](#).
Selection and peer-review under responsibility of the University of Southern California.

Keywords: Large-scale engineering programs, program front-end, complexity, planning, planning mistakes

* Corresponding author. Tel.: +1-617-258-7773
E-mail address: erebenti@mit.edu

1. Introduction

Large engineering programs – also referred to as highly complex or major projects – consist of various projects and sub-projects that contribute to an overall program benefit. These programs include many interrelated components, subsystems and technologies, commonly are socially, technologically and commercially innovative and involve a division of labor among co-specialized organizations that form a (new) network of participants.¹ Examples for such programs can be sociotechnical infrastructure constructions of airports and railways, plant constructions, the development of radically innovative, high-tech industrial products such as electric vehicles or nanotechnology materials; or space research programs.¹ These major projects and programs are affected by a number of serious risks, and subsequently commonly suffer from large cost overruns.²⁻⁶ This study intends to clarify the importance of up-front planning to improve program performance, to propose a model for the front-end of large-scale engineering programs based on a review of existing, suitable models in literature and to better understand the complexity drivers that are impeding reliable planning and the common planning mistakes made in large-scale engineering programs.

2. Overview and Methodology

The interviews were performed to clarify the complexity and challenges in the front-end of large engineering programs and to identify major mistakes observed in the reality of large engineering programs. The interview partners were five project and program planning experts with at least twenty years of work experience from an aerospace and defence company, an aerospace and defence supplier, a consulting firm (2 interview partners) specialized on the planning of large-scale development programs in Automotive, Aerospace and IT, and a multinational engineering and electronics conglomerate. For all interviews, an interview guideline was prepared in advance that included a collection of questions to be discussed and a short introduction into the overall research goals, the front-end model and the current collection of complexity drivers and mistakes. During the one- to two-hours one-on-one interviews, the SMEs were asked about the major activities necessary in the early stages of programs, why they think planning is so difficult for large engineering programs, which mistakes are commonly made, and important factors for successful planning. After each interview, protocols were elaborated to capture the findings. The interviews were performed in the course of three months and supported by continuing literature research to expand the collection of complexity drivers and common mistakes in the planning of large-scale engineering programs. To structure the identified complexity drivers in section 6.1, the categories of complexity of NASA's large space research programs defined by Hoffman & Kogut (technical, organizational, and strategic dimension⁷) were enhanced with two additional categories during the course of the interviews: people and the program environment (compare figure 2). After a review of the interview protocols, major themes of mistakes in the program planning were identified and summarized in section 6.2, also referring to literature recommendations given by the SMEs.

3. Importance of the front-end of programs

An analysis of the reasons for failure in engineering programs identified by Kinscher shows that the majority of errors made in the management of these programs can be traced back to the front-end of programs.⁸ These failures can be too little customer and stakeholder interaction, too little updating on estimated costs during early phases, insufficient probabilistic cost estimations, unrealistic program schedule, no buffers between subprojects, unrealistic staffing ramp-up and ramp-down plan, insufficient resource planning, lacking ability to understand uncertainty and risks, competing resource requirements from different programs, unclear requirement clarification and insufficient multi-attribute tradeoff / trade space exploration. Furthermore previous interviews with SMEs have shown that they consider the early program management activities of scoping, planning and contracting as highly important and more difficult than the activities during the program execution.⁸

Another analysis on 30 CIA and DoD programs concludes, that “most unsuccessful programs fail at the beginning. The principal causes of cost and schedule growth on these large-scale programs can be traced to several causes related to overzealous advocacy, immature technology, lack of corporate technology maps, requirement instability, ineffective acquisition strategy, unrealistic program baselines, inadequate systems engineering, and

workforce issues”.⁹ Further, the author argues, that “early pre-acquisition activities executed in a rigorous fashion can significantly reduce the risk of cost and schedule growth”. Also NASA in a recent study identified mainly failures in the front-end being responsible for unsuccessful programs: “inadequate requirements definition, unrealistic dependence on unproven technology, annual funding instability, complex organizational structures, misapplied cost estimates, scope additions due to requirements creep, acquisition strategy not promoting cost containment”¹⁰. Thus, recently NASA has put emphasis on the program front-end and established a formal planning process within NASA to improve the enunciation of long-term goals and the layout of programs and institutional, and financial plans for meeting the long-term goals¹¹.

In addition, Wirthlin states that existing literature on the front end of commercial product development generally does not consider “large, capital intensive, complex, and technologically advanced products”.¹² In accordance to that, Merrow argues that despite many thousands of pages written on project management, “very little of the literature addresses the peculiar nature of very large and complex projects as a class”¹³. Merrow explores concepts, strategies and practices for successfully planning industrial megaprojects (his definition of these megaprojects is coherent with the definition of an engineering program in this paper) and gives insight into benchmark data of studied projects proving the positive correlation of “Front End Loading” and project performance.

Another case study based analysis¹⁴ exploring the front end of complex projects based on the consideration of nine organizations that are either “world-class or otherwise highly successful in their area of operation” concludes that there is evidence for a managed front end process albeit that process is individual to the sector. The authors also see a gap of literature that supports the understanding of planning of complex projects.

4. Literature review on suitable models to describe the front-end of large engineering programs

In this section, relevant and well-known models from the disciplines of “New Product Development”, “Program Management” and “Management of Major Projects”, and “Systems Engineering” were analyzed to clarify and propose a model for the front-end of large-scale engineering programs. The discipline of new product development was the first to coin the term of a “fuzzy front-end” that precedes a well-structured full-scale development process. The consideration of program or project front-end started more recently and the terminology is still not fixed and clarified.^{14, 15}

4.1. New product development front-end models

One of the earliest models of the front-end of new product development is the three stage front-end model of Smith¹⁶, consisting of the stages “project proposal”, “business plan”, and “detailed project plans & product specifications”. Five years later, Paul¹⁷ suggested a three step model consisting of “idea screen”, “concept development and testing” and “business analysis”. Khurana et al.¹⁸ define a front-end model including a “pre-phase zero”, a “phase zero” and a “phase one”. They describe the pre-phase zero as preliminary opportunity identification and market and technology assessment. Concurrently, the product and portfolio strategy is evaluated. Phase zero defines the product concept, phase one ensures product feasibility and plans the project.

Cooper’s recent version of “Winning at new Products” introduces the “next-generation” stage-gate product development process.¹⁹ Further, he introduces three versions of the process: “Full”, “Xpress” and “Lite” and explains that depending on the involved risks, the number of gates to be run through differs. According to Cooper the front-end or “predevelopment stages” start after the generation of an idea and (for high-risk and innovative projects) include “scoping” and the “building of a business case”. For technology developments and technology platform projects “which spawn multiple new-product projects”, Cooper in addition introduces a three-stage technology development process. After having passed the fourth gate of the technology development process, these projects enter at the second gate of the new product process. Wirthlin¹² analyzed different front-end processes based on case studies and came up with a four-step framework including the steps “identification of requirements”, “initial screening”, “concept development”, and “business case development”. “Proving clarity and a common language to the “fuzzy front end”, Koen et. al.²⁰ describe the front end of new product development using five non-sequential activities and imply the iterative nature of activities in the front end: “idea genesis”, “idea selection”, “concept & technology development”, “opportunity analysis”, and “opportunity identification”. Further, they describe the nature

of work in the front-end as “experimental, often chaotic” and “difficult to plan”. Ulrich and Eppinger introduced a generic five-phased product development process and recently added a sixth phase, the “planning” phase in the fifth edition of “Product Design and Development” that precedes the project approval and launch of the actual product development process including the activities of opportunity identification, assessment of technology developments and market objectives.²¹ The following phase of “concept development”, also referred to as “front-end process” by the authors, includes the identification of customer needs, the establishment of target specifications, the generation and selection of product concepts, testing, setting of final specifications and the planning of downstream development.

4.2. Early stages in Systems Engineering

The International Council for Systems Engineering describes the life-cycles stages of a technical system with seven phases, including the first phases of “exploratory research” and “concept” through the last phase of “retirement”.²² The mentioned first two phases can be considered as parts the front-end of large-scale engineering programs. The phase “exploratory research” includes the identification of stakeholder needs and the exploration of technologies and ideas. The “concept” phase comprises the refinement of stakeholder’s needs, the exploration of feasible concepts, and the proposal of a viable solution. INCOSE²² further describes that the phases are separated by decision gates with the options of proceeding to the next stage, continuing in the current stage, returning to preceding stage, putting a hold on a project, or even terminating it.

4.3. Early stages in Program Management

The Project Management Institute (PMI) recently published the third edition of its Standard for Program Management explaining the program front-end, the “program definition” phase to consist of two phases: “program formulation” and “program preparation”. The involved activities during these phases comprise the scope planning, stakeholder identification and engagement, communication planning, cost and schedule estimation, funding, procurement planning, quality planning, resource planning, and risk management planning. The Office of Government Commerce (OGC) offers another program management standard, mainly focusing on managing change in organizations. The first two phases of programs according to OGC are “identifying a program” and “defining a program”. The identification phase includes the activities of sponsoring, confirming the program mandate, the developing of a program brief that documents the desired benefits and the commitment to investment and resources, and the appointment of key responsibilities. The definition phase comprises the activities of establishing the program team and infrastructure, planning the organization and governance arrangements, identifying stakeholders, refining the program vision, developing of blueprint, development and modeling of benefit profiles, identification of tranches (temporal groups of activities and projects), program planning, developing and confirming of a business plan, and the preparation of the first tranche.

4.4. Major project Management front-end models

Archibald et al. argue that current project management standards are missing a sufficient front-end description. They introduce an “incubation phase” in which an “embryonic knowledge and understanding about the project” is built up through a process of information accumulation.¹⁵ Samset introduces an idealized front-end model of technocratic decision-making upfront in projects arguing that in reality the decision making process may be complex, unstructured and affected by chance and analyses may be biased or inadequate.²³ Edkins et al. define the front-end of a project as the “preliminary emergence phase” beginning with the authorization of the management to spend time, money and effort to commence development of the project definition. The front-end finishes either with acceptance by sponsors, the termination or shelving of the project at a sanction gate. Suitable models for specifically the front-end of engineering program front-end exist that focus on very specific industries and types of major projects. Merrow¹³ suggests a three phase front-end loading model for oil and gas, chemicals and minerals megaprojects consisting of the phases “appraise opportunity”, “develop scope”, and “define project”. Recently, Wessen et al. have developed and implemented a method for measuring and communicating the maturity of space

mission concepts at NASA describing eight maturity levels from the initial “cocktail napkin”, over the levels of “initial feasibility”, “trade space”, “point design”, “baseline concept”, “integrated concept”, preliminary implementation baseline” to level eight, the “project baseline”.¹¹ They further stress the iterative nature of work between “initial feasibility”, “trade space” and “point design”. The eight new concept maturity levels at NASA were developed “to provide structure in Pre-Phase A where none existed before.” Further its purpose is to offer mission architects a scale to communicate the amount of progress incorporated into a mission concept, which allows the architects to know how much design work is needed to mature a concept to Phase A, The Concept & Technology Phase.¹¹

5. Proposing a front-end model for engineering programs

For the purpose of this study, the program front-end process is defined that starts with the identification of an opportunity from strategic considerations and the authorization of sponsoring stakeholder(s) to put time and effort into the development of a program plan. It ends with a final sanction gate, after which the program will either be fully funded, paused or terminated, and major contracts with suppliers, partners and customers are signed.

The main purpose of the front-end activities is to reduce uncertainty and increase knowledge in the three knowledge domains of the product concept, the underlying business plan and necessary organization and processes to deliver the program’s product. Through an integrated exploration of different options for product concept, organizational structures and processes and their impact on the business plan, the program plan is matured and the program organization becomes more aware of advantages and disadvantages of different options and thus more resilient to changes caused by uncertainty during the program life cycle that is unforeseeable at the program start. Most of the considered models suggest a gated process, often including a definition of program plan artifacts that have to be completed by the specific gates (Cooper, Ulrich and Eppinger, Mellow, INCOSE, PMI, Wessen). According to the SMEs (detailed explanation of interview results in section 6) a gated processes is indeed necessary for large-scale engineering programs, but not sufficient as the gated process may lead to slow decision speed and can induce too much bureaucracy and a compliance mentality preventing sufficient reflection and thinking by participants. Consistent with Cooper¹⁹, the SMEs interviewed encouraged fast learning cycles and the integration of spiral and agile development elements into the stages to enable fast knowledge exchange as especially in the early stages a lot new knowledge is gathered in different disciplines. The following three paragraphs, structured by the knowledge domains, give an overview of the front-end activities that were identified during the literature review. The front-end activities are performed simultaneously in an iterative nature. With increasing progress the number of options are decreasing and the level of elaborated information about product concept, business plan and organization and processes is increasing.

Product concept. The knowledge about the product concept is refined by the activities of user need and program value identification, the deployment of requirements, the identification of solutions and the assessment of these solutions and related technologies, the architecting and integration of the overall system and a validation and verification (at least of most critical program components). Throughout the process the identification of technical risks is performed.

Business plan. The knowledge about the business plan is increased through the activities of exploring the program’s alignment with the overall strategy, a market assessment (for market-driven programs), clarification of sponsoring stakeholders’ budget limitations (for customer-driven programs), the program cost and schedule estimation, funding and the identification of external risks and sources of uncertainty.

Program organization and processes. To refine the knowledge about the necessary program organization and processes to deliver the program’s product the following activities are performed: The planning of activities, the identification of critical interdependencies of activities and handoffs, the planning of communication in the organization, the setup of a program governance, the engagement of stakeholders (customers, suppliers, sales/distribution, support, regulatory, etc.), contracting with key suppliers and partners, the engagement of manufacturing, staffing and staff planning, resource planning, the identification of organizational risks and the application of lessons learned from previous programs.

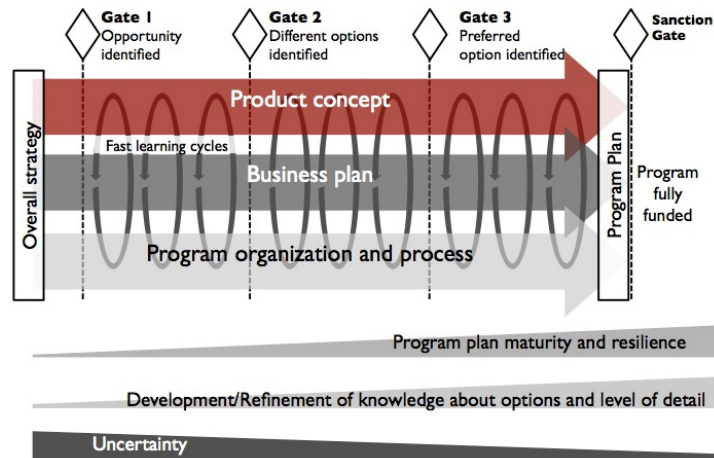


Fig. 1. Suggested model for the front-end of large-scale engineering programs

6. Discussion of results from interviews with subject matter experts

6.1. Complexity drivers preventing reliable planning in the front end of engineering programs

To be able to optimize the activities in the front end of large-scale engineering programs, it is important to understand the factors that make planning of programs difficult and increase the level of uncertainty. Five categories are used to structure the complexity drivers: environmental complexity drivers and program internal drivers with the sub-category of technology, people, organization and process and strategy. The following paragraphs explain the identified areas and complexity drivers.

Technology. The technical factors that make planning difficult are the high number and various types of interfaces of components, and interdependencies among technologies. Furthermore, in order to achieve NASA's mission goals - or as a technology company - to remain competitive in highly competitive markets, technological advances are required. However, aggressive technology advances introduce further uncertainty into program planning and programming.²⁴ In addition, through technical integration of already complex subsystems in larger and larger systems or software and hardware a change in one place leads to thousands of complications in others – driving the need for further coordination and impeding the predictability of development efforts.

Organization and process. The organizational factors that drive complexity and complicate planning are a higher number and variety of partners from industry and academia in a more globalized business environment, distributed and virtual teams, decentralized authority, and intensive learning needs due to the unique character of the programs.^{7, 25} Often these teams are working together the first time, which is why not only learning regarding the program deliverables themselves but also regarding the social interaction with new partners is required.²⁶ Furthermore, often no management structure is in place from the start as the organization is set up together with the program itself. Thus, roles and responsibilities are not clarified and decision-making processes, especially in the early phases, are slow. According to one SME, newly-defined program organizations often lack the personal relationships – internally and externally, e.g. to the suppliers - that are important for an effective and efficient knowledge exchange in the planning phase. Another complexity driver impeding sustainable planning is the staff turnover coming along with long program life cycles – if an important member of the planning team leaves the program, tacit knowledge will get lost and the planning benefits decrease. Also, a program itself involves many different subprograms and projects, boundaries between projects, combination of projects and interplay of project life cycles, generating additional complexity.¹ Another organizational factor mentioned by a SME that enforces insufficient planning can be a wrong incentive structure that pushes program managers to generate "deliverables" as soon as possible as their success is measured based on completed program milestones instead of reduced levels of uncertainty to an acceptable level before taking action.

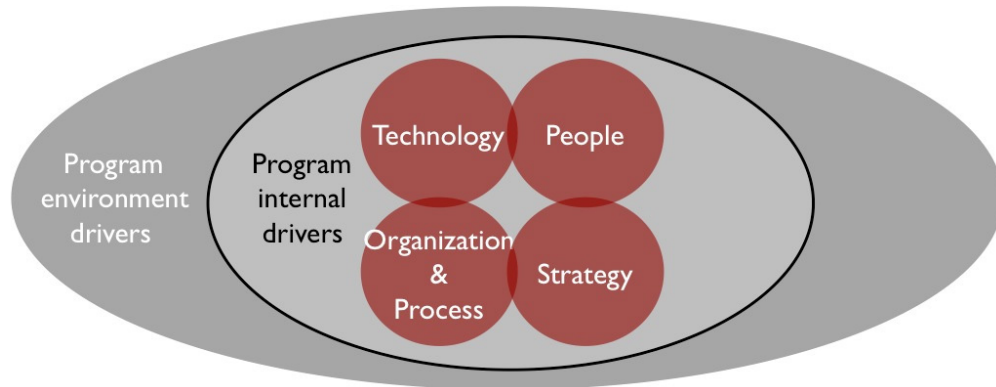


Fig. 2. Categories of complexity driver impeding reliable planning in large-scale engineering programs

Strategy. The strategic factors that drive complexity and planning constraints are a high number and diversity of stakeholders, often decentralized decision authorities, a socio-political context of the programs, complexity of funding sources and processes and competing geopolitical interests in international partnerships. Furthermore, sponsor stakeholders especially in the early program phases are risk averse and afraid of funding an exhaustive process to explore different alternatives and the trade-space. Additional strategic factors preventing sufficient planning mentioned by the SMEs can be: insufficient time until the proposal deadline for the customer or scattered responsibility for concept development and budget and costs.

People. The human barriers to sufficient planning comprise limited human cognition and the ability to deal with complex decisions, insufficient planning know-how and experience with complex and far-reaching decisions or various intentions of program participants who also pursue personal interests or the interest of their actual organization that might not always be aligned with benefits for the program. This can lead to hidden scope in planning and contracts or to “low-balling” of cost and schedule estimates through the sales team that tries to sell a contract by all means. Furthermore, as already mentioned the program teams often are working together for the first time, which requires not only learning regarding the product itself but also regarding the social interaction with new partners and can be made even more difficult by different cultural or professional backgrounds. Also, the tendency of people to include buffers in personal estimations of task durations coupled with the “student syndrome” to start tasks at the latest possible point in time increases planning inaccuracy.

Environment. Furthermore, the changing program environment over the program life cycle increases the complexity of program planning efforts. Programs with long life cycles of often more than five years have to cope with a changing environment⁷: New, more advanced technologies suddenly can be available after the product architecture and technologies to be applied were already defined. Customer needs can be influenced by these new technologies or by unforeseeable regulatory changes. Also, global supply chains may have changed, key suppliers may disappear from the market or better suppliers might appear after contracts were already signed with their competitors. Raw material prices may have changed against those assumed for development and increase program costs.

6.2. Common mistakes during early phases in the front end of engineering programs

The interviews with SMEs were used to also collect and understand major mistakes in the planning process of large engineering programs and projects. All the subject matter experts interviewed in this study agreed on the fact that planning of large-scale engineering program still remains one of the biggest challenges in today's programs and that the current planning methodology needs to be improved. The following section gives an overview of the identified themes of common mistakes in the front-end according to the SMEs, also referring to literature recommendations given by the SMEs.

Missing efforts to ensure everyone is on the same page. According to one SME the first step in a program should be to make sure that everyone is on the same page as far as the customer value proposition is concerned, but often this is not done adequately: The program managers may think that they know the customer value but this knowledge is not communicated throughout the program.

Up-front efforts considered as costs, not as investment. Program managers often do not perceive the value of up-front planning as an investment that helps to reduce the amount of rework, but rather see planning efforts as additional costs.

Neglecting fast learning cycles. According to the SMEs integrating planning events like well-prepared planning workshops often are not taken seriously enough and organizations rush through them without putting enough rigor into them and without offering trained facilitators that are capable of asking critical questions about the plan maturity. Senior management and necessary experts often miss those integration events or delegate the participation to someone else less experienced. This way cross-functional risks may not be identified.

Letting uncertainty flow downstream. Another major mistake listed by the SMEs in the planning phases is postponing the clarification of reducible uncertainty to a later point in time and not focusing resources on the aspects first that are most uncertain. This “We will figure that out later” mentality carries uncertainty to the next stages of the program and induces unnecessary risks. Often program managers let programs go forward to the next stage without having clarified TRLs or resource availabilities so as to “not hold up the program” – and undermine the purpose of the gated process.

Too detailed planning too early and over-specifying. According to the SMEs too much effort and detail too early in the front end will likely force reliance on more assumptions than is prudent, resulting in rework and a decrease in performance later. One of the SMEs explains that often early in the programs he participated in, program plans in tracking documents already included thousands of activities, although the actual knowledge about what had to be done was not available yet. According to the SME, programs then were managed based on these guesses of what the task durations and milestones were. In the end, forcing people to be specific about their task duration caused rework in the planning process itself (with high effort to update the excessively-large program plan) and wrong expectations of sponsors and stakeholders.

Neglecting communication and coordination efforts. While putting too much effort on planning program details, other information like the obvious need for communication and coordination efforts, or different time zones within the program organization that can be known early on, are not taken into account when defining necessary activities and estimating the schedule. According to the SMEs, communication and coordination efforts are especially important in large-scale engineering programs with complicated products and outcomes that require a large organization with divided competencies of program managers and system engineers to ensure that everyone is aware of the impact of his actions on others and the environment he is working in. The use of Gantt charts and other simplified project plan representations that do not show the necessary coordination between often thousands of program participants²⁵ supports this mistake.

Neglecting organizational setup and optimizing processes. One of the SMEs considered his organization much more capable in managing technical risk than in coping with organizational risks. The company tends to spend a lot more time on resolving the technical risks, neglecting the identification of resource bottlenecks and setting up appropriate organizational structures and processes to ensure the coordination between all people involved in the program – resulting in programs in the company more often failing due to organizational risks than technical risks.

Using concrete assumptions rather than ranges including uncertainty. Instead of keeping track of uncertainty and defining requirements in ranges (e.g. 30-90°C), expected averages and concrete values (e.g. 60°C) are assumed and formalized in the requirements that later might turn out to be wrong and cause rework.

Neglecting different mental models. The presence of different cultures, mentalities, professional backgrounds, etc., may lead to differing mental models in global programs. Thus, a missing common understanding of how long certain activities will take and how much buffer is put into personal activities can lead to wrong embedded assumptions in the overall program plan. In addition, missing common behaviors can result in inefficient communication efforts.

Planning as a one-time effort. Another mistake identified in the interviews is the assumption of planning as an activity performed once at the beginning of a program. This results in a detailed program plan that is used in the

follow up for evaluation, even though the plan may be outdated. Often, the inability to quickly update a plan if program aspects change is a driver for this mistake.

Using complexity of programs as an excuse for low TRLs. According to one SME the complexity of a technical product is often used as an excuse for the selection of technologies with a low TRL. The selection of a technology with low TRL is often an indicator of not having had a coordinated strategy for exploring the technologies that are needed on the next program and not having spent enough time, resources and R&D efforts in advance.

Focus on a point-design too early. Another mistake observed by the SMEs in the early stages in programs was the focus on one solution too early or starting with a predefined solution (that might have been the favorite solution of, e.g., senior management) without having explored other options that best fulfill all the requirements. As a result, at advanced gates in the stage-gate process the programs tend to be stopped or reverted to previous stages, causing tremendous rework. The evaluation of different technical options alone is not enough to overcome this. Different program plan options may not be discussed and evaluated (driven perhaps by insufficient planning tools that do not enable quickly comparing program plan options in the dimensions of related risks, costs or schedule), which leads to the fact that “often early mission studies focus on product/ mission desirability yet don’t sufficiently include feasibility from a project management view”.²⁷

Reliance on compliance and checkboxes. The reliance on compliance and checkboxes in checklists according to one SME in his company prevented people from taking responsibility for their decisions, as they could always blame the process if something went wrong. What is needed is a knowledge mentality and the initiative to think about the relevant questions proactively to improve the understanding of the product and the program. Instead, certain program plan artifacts may be generated to satisfy compliance with internal processes. Another SME notes that it is not enough to have a filled program template including all required program artifacts. It is far more important to consider how the program plan was developed (Was everyone involved and did everyone give his buy-in? Were different options considered?) in order to be able to appraise its maturity and reliability.

7. Conclusion

Considering a lack of models that offer an integrated view of program management, product development and systems engineering activities in early engineering program phases, this study developed a basic model to describe the front-end of large-scale engineering programs based on a broad literature review. Additionally, interviews with subject matter experts identified and documented complexity drivers that impede reliable planning and major common mistakes in the program front-end. The mistakes identified and further observations of successful industry practices in the program front-end can be used to deploy success factors and to develop more detailed guidance for practitioners.

References

1. Project Management Association of Japan, *P2M Project & Program Management for Enterprise Innovation - Guidebook New Edition*. Vol. 1. 2008.
2. GAO, *Defense Acquisitions - Managing Risk to Achieve Better Outcomes*, in *GAO-10- 374T*2010, United States Government Accountability Office: Washington, D.C.
3. Tang, C.S. and J.D. Zimmermann, *Managing New Product Development and Supply Chain Risks: The Boeing 787 Case*. Supply Chain Forum: An International Journal 2009. **10**(2): p. 74-85.
4. Michaels, D., *Airbus Woes Darken Jet Delivery*. Wall Street Journal, 2010.
5. Spiegel Online International, *Cost Explosion: Price Tag for New Berlin Airport Keeps Rising*. Spiegel Online International, 2013.
6. Wassermann, A., *ECB Headquarters: Skyrocketing Costs for Skyscraper Project*, in *Spiegel Online International*2013.
7. Hoffman, E.J. and M. Kohut, *NASA’s journey to project management excellence*. National Aeronautics and Space Administration, 2013.
8. Kinscher, K., *Identification of Lean Enablers for Program Management*, in *Lean Advancement Initiative (LAI)*2011, MIT: Cambridge.
9. Meier, S.R., *Best project management and systems engineering practices in the preacquisition phase for federal intelligence and defense agencies*. Project Management Journal, 2008. **39**(1): p. 59-71.
10. NASA, *Program and Project Improvement Initiatives*. Knowledge Magazine, 2007. **Spring 2007**: p. 50-54.
11. Wessen, R.R., et al., *Space Mission Concept Development Using Concept Maturity Levels*. AIAA SPACE 2013 Conference and Exposition, 2013.
12. Wirthlin, J.R., *Best Practices in uses needs/requirements generation*, in *Center for Technology, Policy and Industrial*

- Development 2000, Massachusetts Institute of Technology: Cambridge, MA.
13. Merrow, E.W., *Industrial Megaprojects - Concepts, Strategies, and Practices for Success* 2011, Hoboken, New Jersey: John Wiley & Sons, Inc.
14. Edkins, A., et al., *Exploring the front-end of project management*, in *Engineering Project Organizations Conference* 2012: Rheden, NL.
15. Archibald, R.D., I. Di Filippo, and D. Di Filippo, *The Six - Phase Comprehensive Project Life Cycle Model Including the Project Incubation/Feasibility Phase and the Post - Project Evaluation Phase* PM World Journal, 2012. **1**(5).
16. Smith, P.G. and D.G. Reinertsen, *Developing Products in Half the Time*. 1 ed 1991: John Wiley & Sons.
17. Paul, R.N., *Evaluating Ideas and Concepts for New Business-To-Business Products*, in *The PDMA Handbook of New Product Development*, M.D. Rosenau, et al., Editors. 1996, John Wiley & Son, Inc.
18. Khurana, A. and S.R. Rosenthal, *Towards Holistic "Front Ends" in New Product Development*. Journal of Product Innovation Management, 1998. **15**(1): p. 57-74.
19. Cooper, R.G., *Winning at new products : accelerating the process from idea to launch*. 4 ed 2011, New York: Perseus Book Group.
20. Koen, P., et al., *Providing Clarity and a Common Language to the "Fuzzy Front End"*. Research-Technology Management, 2001. **44**(2): p. 46-55.
21. Ulrich, K.T. and S.D. Eppinger, *Product Design and Development*. 5 ed 2012, New York: McGraw-Hill.
22. INCOSE, *INCOSE Systems Engineering Handbook* ed. S.H.W.G.a.I.C.o.S. Engineering. Vol. v. 3.2.2. 2011.
23. Samset, K., *Projects, Their Quality at Entry and Challenges in the Front-end Phase*, in *Making Essential Choices with Scant Information - Front-End Decision Making in Major Projects*, T.M. Williams, K. Samset, and K.J. Sunnevåg, Editors. 2009, Palgrave Macmillan: New York. p. 18-35.
24. Rebentisch, E., *Preliminary Observations on Program Instability*, , in *LAI Whitepaper* 1996: Cambridge, MA.
25. Moser, B. and J. Halpin. *Virtual Teams - The Design of Architecture and Coordination for Realistic Performance and Shared Awareness*. in *PMI - 2009 Global Congress*. 2009.
26. Moser, B., *Projects as Socio-Technical Systems: Observation, Forecasting, and Design of Communication Activity*, 2013: Rocky Mountain Project Management Symposium.
27. Moser, B., W. Grossmann, and P. Murray. *Simulation & Visualization of Performance across Subsystems in Complex Aerospace Projects*. in *PMI 2009 Global Congress Proceedings*. 2009.